

King Abdullah University of Science and Technology

KAUST Academy – CEMSE Division

MPS Mol-AI and MPS Mol-Cyber Security Bridging Program: Python Syllabus

Course dates JAN 5-9

Course Objective

The objective of this Python programming course is to provide beginners with a comprehensive introduction to the core concepts and tools needed to write simple, functional Python code. By the end of the week, students will have the foundational knowledge necessary to begin writing their own scripts and tackling basic programming challenges.

General Description

The course begins with an introduction to Python on Day 1, where students are familiarized with the language's history, key features, and installation process. On Day 2, the focus shifts to understanding Python's syntax, variable types, and basic operations, along with type conversions. Day 3 explores Python's essential data structures, including lists, sets, tuples, and dictionaries, while providing students with the skills to manipulate these data types. On Day 4, students will learn how to use conditional statements and Boolean expressions to introduce decision-making into their programs. After completing these four days, students will be prepared for the course exam on Day 5, which includes a 30-minute quiz in the morning, followed by a 120-minute programming test in the afternoon. The exam will assess their ability to apply the concepts learned during the course by writing Python code, and students will be allowed to refer to course slides but not use the internet during the test.

Instructors

The instructor for the course is Rama AYOUB (rama.ayoub@kaust.edu.sa). The three teaching assistants are Rafael Menaca Cabrera, Hikmatullo Ismatov and Victor Ceballos Inza.

Schedule

The class meets for a week (Sunday to Thursday). Each day consists of two different types of sessions: lectures and problem solving sessions. During each lecture, the topic is presented by the instructor. After the lectures, the students solve problems in small groups to apply, practice, discuss and reinforce their learning. Both the instructor and the teaching assistants are available to help and answer questions.

Detailed Plan

The plan for the week is as follows:

Sunday: Introduction to Python

- **What is Python?**
 - Overview of Python: The history and What makes Python a popular programming language?
 - Python's uses in various fields (web development, data science, automation, etc.)
 - Advantages of Python (easy to read, versatile, large community support)
- **How to Install Python?**
 - Installing Python on Windows, macOS, and Linux
 - Introduction to running Python scripts from the command line

Monday: Python Syntax & Basic Data Types

- **Syntax and Indentation**
 - Python's clean syntax and importance of indentation for block structures
 - Writing simple Python statements and expressions
 - How to avoid common syntax errors
- **Comments**
 - Single-line comments
 - Multi-line comments
- **Variables and Basic Commands on Variables**
 - Creating variables and assigning values
 - Understanding different data types: integers, floats, strings,...
 - Using Python's built-in functions (e.g., `print()`, `type()`, `input()`)
- **Conversions Between Different Base Types**

- Type conversion: `int()`, `float()`, `str()`
- Converting data between types (e.g., string to integer, float to string)

Tuesday: Working with Data Structures

- **Lists**

- Creating and accessing elements in lists
- Common operations: `append`, `remove`, `pop`, `slice`

- **Sets**

- What is a set? Differences between lists and sets
- Adding, removing, and checking membership in sets
- Set operations: `union`, `intersection`, `difference`

- **Tuples**

- Defining and using tuples (immutable sequences)
- Tuple unpacking
- When to use tuples vs lists

- **Dictionaries**

- Creating dictionaries (key-value pairs)
- Accessing, adding, and modifying dictionary entries

Wednesday: Decision Making and Boolean Expressions

- **Decision Making: If Statements**

- Structure of `if` statements: Simple and compound conditions
- Using `else` and `elif` for multiple conditions

- **Boolean Expressions**

- Understanding Boolean values (`True` and `False`)
- Logical operators: `and`, `or`, `not`
- Comparing values with relational operators (`==`, `!=`, `>`, `<`, `>=`, `<=`)

- **Examples and Practice**

- Writing programs that make decisions based on conditions (e.g., checking if a number is positive, negative, or zero)
- Nested `if` statements and complex conditions

Thursday

- Morning:
 - Review session of Key Concepts
 - Answering any questions
 - MCQ test
- Afternoon:
 - Test

Evaluation

On the final day, after the review and Q&A session, students will take a 30-minute quiz in the morning, consisting of 10 multiple-choice questions (MCQs). This quiz will be a closed-book, closed-notes assessment, with no calculators or laptops allowed. In the afternoon, students will complete a 120-minute programming test to assess their ability to write Python code. The test will be conducted on a laptop with no internet access, and students may only refer to the course slides.

- 10 multiple choice questions.
- 3 questions where the student implement their own codes.